Evaluation of Spatial Filtering Algorithms for Visual Interactions in CAVEs

Antoine Lassagne
Renault – Arts & Métiers

Andras Kemeny
Renault – Arts & Métiers

Javier Posselt
Renault

Frédéric Merienne
Arts & Métiers

***Abstract*—We present an approach to solving the problem of haptic and visual misalignment in CAVEs. The approach moves the collision box for the virtual screen's buttons to coincide with where the user perceives their virtual location. Different filtering strategies were used. We evaluated the algorithms with simulations and with real subjects.**

The recent improvements in high performance immersive systems, mostly achieved with enhancements in visual systems, bring a flow of new applications of virtual reality in the industry. At Renault, these new uses bring specific requirements in terms of interactions, as engineers aim to work on virtual car prototypes with their regular methods (transformations, explorations, measurements, cut planes, etc.) in immersive systems. However, strong visual/haptic mismatches prevent users from effectively using interaction modalities.[1] Due to these mismatches, the visual and the haptic workfield are not collocated and the users do not see what they touch where they touch. Hence, few interaction methods are actually implemented in these systems.

In this paper, we evaluate the use of different filtering strategies meant to determine where the users try to touch buttons. The data collected allows the system to move the collision box of the buttons according to the previous interactions. Such a system would increase the precision of the user's interactions and reduce their mistakes. We optimized several algorithms and performed an experiment to validate our findings.

RELATED WORK

Visual Match

The CAVE itself has some particularities that must be considered when implementing the correct interaction modality. Unlike in head-mounted displays, the users see their own body (and hands) in a CAVE. This should be an advantage for natural interaction, but it is also a source of problems.

The event of a touch is binary whereas the visual perception of depth is continuous; it is hence important that visual and haptic contacts are matched. However, seeing objects too far or too close while touching them is a common complaint from CAVE's users. There are known causes, like the mismatch between the tracked three-dimensional (3-D) glasses that are used to calculate the frustrums from the eyes' point of view and the actual eyes. Different interpupillary distances, eye depths and nose heights lead to an offset between the theoretical and the real eyes, resulting in a different spatial perception between users.

It has also been established that distance and scale perception can be inaccurate in CAVEs, depending on the quantity of objects of known size, the photorealism of the scene, the quality of the visual system, and even the duration of the simulation.[5] The brain, lacking visual cues to get a robust spatial perception, can give unreliable information, resulting in drifts.[6] Research has shown that these drifts of human perception can be reduced with interaction, tasks, and context,[7,8] but little is known about this topic as of yet.

These issues are significant, as there is a low probability that the users see their hand and the virtual point they want to touch at the same location.[9]

Interaction Modalities

Cave Automatic Virtual Environments (CAVEs) are immersive virtual reality systems where images are projected on the walls of a room-sized cube. Multiple research works evaluated different interaction systems in CAVEs. The key of an effective interaction is relevant feedback that can be attained through software and/or hardware. Several experiments demonstrated that kinesthetic feedback,[2] cutaneous feedback,[3] and sensory substitution[4] can all significantly enhance the interaction capabilities of virtual environments.

However, these modalities also have their own flaws, depending on their type: price, bulkiness, software/hardware compatibility, ergonomics, or even capabilities. Kinesthetic systems are especially expansive and unhandy. Sensory substitution is the easiest solution to implement, as it only requires a software solution and a tracking system. It does not prevent users from passing through objects. Nevertheless, visual cues and sounds can be relevant and help people in their interactions. This study is focused on sensory substitution for its higher compatibility with CAVEs and its flexible usage.

Existing Solutions

There exist a few methods to handle the mismatch between the haptic and the visual workspace, like the clutching, scaling, and bubble interaction techniques.[10] However, these methods are often designed to enlarge the range of interaction devices, not to colocate the haptic and the visual workspace.

Many CAVEs do not handle visual drift. They settle for inaccurate interactions, no interactions at all, or interactions with a virtual tool. Virtual tools, displayed by the CAVE, encounter the same drift as the rest of the virtual environment. They are thus easier to interact with, as users easily immerse themselves into avatars not sharing the same location (computer mice are a good example), and interactions are efficient with them.[11,12]

In other fields, solutions exist to improve the performance of inaccurate interactions. For example, the patent FR3028968B1 describes a car human-machine interface (HMI) system that predicts the intention of users to ease their interactions by moving the buttons. Likewise, developers of smartphones' virtual keyboards use tricks to improve performance: although the process is invisible, the size of keys are changing after every tap, on the grounds of probabilities and known dictionaries.

These two solutions are transparently changing the collision boxes of the buttons to adapt to the user's errors, and this is what we aim to reproduce in CAVEs.

OBJECTIVES AND HYPOTHESIS

Scope of This Study

The HMI designers of car dashboard touchscreen software currently test their creations on desktop tablets. They should be able to evaluate

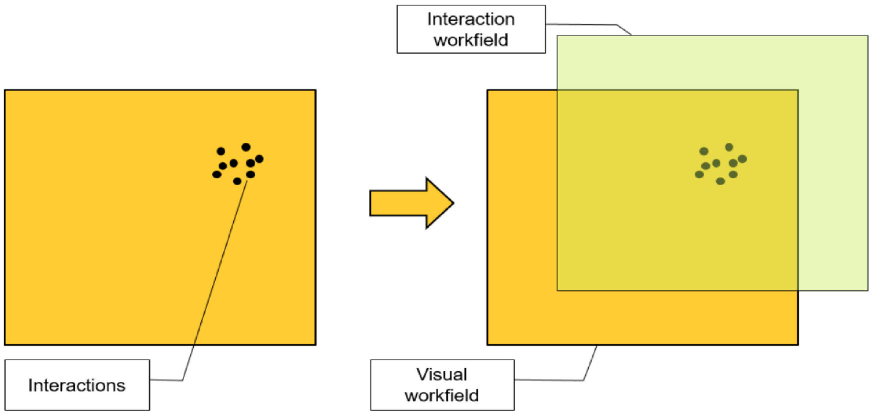

Figure 1. The collision box of the button is moving regardless of its visual.

them easily in driving conditions, in the correct vehicle, without building a new prototype every time. Virtual reality offers the flexibility and convenience that they seek, on the condition that the interaction modalities reach sufficient performance and reliability.

This research is the third part of a series of studies carried out to evaluate the performance of interactions in CAVEs for HMI design purposes.[6,7] The interactions are limited to a bidimensional plane representing the dashboard touchscreen of a virtual car. Handling only two dimensions allows easier design and evaluation of the interaction system, before porting it to tridimensional use-cases.

We aim to reduce the visual mismatch discussed earlier, as it is often responsible for the poor interaction experience. Following the example of what exists in other fields that also encounter interaction performance issues, we wish to implement an algorithm that can shift the collision box of buttons depending on the drift of the user. Natural interactions with the hand are targeted, as these interactions are desired in industrial usage and they face a significant mismatch between the virtual objects and the user's hand.

Hypothesis

Two major hypotheses are assumed in this experiment.

- We assume that the users unconsciously try to reach the center of the buttons when touching them.[13] In this experiment, subjects were specifically instructed to hit the center of the buttons to make sure that the study would not be biased.
- On the basis of other studies and observations, we assume that there are two kinds of spatial drifts encountered by users.
    - A systemic offset, mainly due to morphologic disparities and system imperfections, depending on the system, the point of view, and posture.[5]
    - An uncertainty offset, due to the brain lacking robust perception cues.[14]

Hence the algorithm has to handle both offsets as well as possible.

BUILDING THE ALGORITHM

How it Should Operate

In technical terms, the algorithm is supposed to move the collision box of the buttons to make it coincide with what the user sees. For example, if a user perceives a button too much on the right compared to the model, the collision box moves to the right, as represented in Figure 1.

The global mechanics of this system require specifications to define the behavior of the algorithm with more precision.

- The visual representation must never move, and the operation is invisible for the user.
- The collision box cannot go too far from its initial position to keep it from being lost.
- The operation must compute in real time.
- The operation must significantly enhance the performance of the interactions and be appreciated by the users.
- The operation must take into account human error.

Different Algorithms

Different strategies were implemented for testing purposes. For each solution, (x_i, y_i) represents the coordinates of the center of the collision box and (x,y) represents the coordinates of the last interaction.

1. Linear filtering – The simplest method, following a linear equation (1).

$$\begin{cases} x_{i+1} = x_i + K(x - x_i) \\ y_{i+1} = y_i + K(y - y_i) \end{cases} \tag{1}$$

2. Quadratic filtering – Not much more complicated, but a different behavior (2).

$$\begin{cases} x_{i+1} = x_i + K * \sqrt[a]{(x - x_i)} \\ y_{i+1} = y_i + K * \sqrt[a]{(y - y_i)} \end{cases} \tag{2}$$

3. Mobile means filtering – This kind of filtering takes more interactions into account, to handle human errors more smoothly (3).

$$\begin{cases} x_{i+1} = \frac{\sum_{k=i-m+1}^{i} (K_k * x_k)}{m} \\ y_{i+1} = \frac{\sum_{k=i-m+1}^{i} (K_k * y_k)}{m} \end{cases} \tag{3}$$

4. Fuzzy logic filtering – A complex algorithm intended to adapt quickly and handle human errors.[15]
5. PID regulators[16] and Kalman filters[17] were considered, but these are adapted to dynamic use cases, whereas our interfaces are not moving within the environment.

The Fuzzy Logic Algorithm

As it provided the best results in simulations (described further), the fuzzy logic algorithm needs more details. It is built on additional specifications:

- It must take into account the past successes and errors from the users.
- It must not consider one isolated missed interaction. One cannot remain focused all the time and to err is human.
- It must rapidly correct if the user suddenly drifts.

To respond to these inquiries, we set up four linguistic variables.

- The proximity to the last interaction
  (A, $[-100, 100]$, {Near, Medium, Far}).

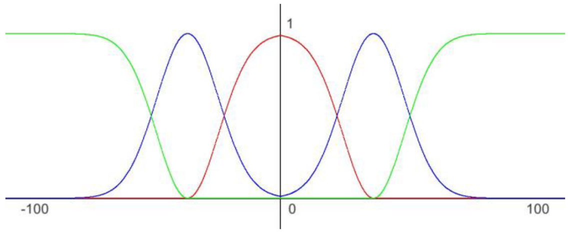

Figure 2. Standard Gaussian functions of fuzzy sets. The centered red function is for Near/Low sets, the blue one is for Medium set, and the fringe green function is for Far/High sets. The equations are $f(t) = \exp \frac{-1}{2} (\frac{t-\mu}{\sigma})^2$.

- The proximity to the last weighted mobile mean
  (B, $[-100, 100]$, {Near, Medium, Far}).
- The correction wanted relative to the last interaction
  (X, $[0, 100]$, {Low, Medium, High}).
- The correction wanted relative to the last weighted mobile mean
  (Y, $[0, 100]$, {Low, Medium, High}).

The fuzzy membership functions are Gaussian functions represented in Figure 2. Their sum is 1 all along $[-100,100]$, their parameters are fixed to $\mu = 50$ and $\boldsymbol{\sigma} = 15$.

We use Zadeh operators as fuzzy operators, a common replacement of basic operators. The fuzzy output function fz depends on the membership of one interaction in the linguistic sets A and B.

fz is the final function, the sum of normalized partial functions (see Equation (4)). The 6 partial functions lead to 6 linear corrections (with gains Ki), getting closer to the last interaction and the weighted mobile mean. The algorithm is thus also subject to the adjustments of the gains of each partial function.

$$f_z = \sum_i f_z(X_i) + \sum_i f_z(Y_i) \tag{4}$$

The matrix of decisions, defined in (5) and (6), is arbitrary and is the key to achieve our objectives. For example, if the current interaction is far from the last one but near the mobile mean, the partial function $f_z(Y_2) \in (A_2 \cap B_0)$ is predominant.

$$\begin{cases} f_z(X_0) \in (A_0 \cap B_0) \cup (A_1 \cap B_0) \cup (A_2 \cap B_0) \\ f_z(X_1) \in (A_0 \cap B_1) \cup (A_1 \cap B_1) \\ f_z(X_2) \in (A_0 \cap B_2) \end{cases} \quad (5)$$

$$\begin{cases} f_z(Y_0) \in (A_0 \cap B_0) \cup (A_0 \cap B_1) \cup (A_0 \cap B_2) \\ f_z(Y_1) \in (A_1 \cap B_0) \cup (A_1 \cap B_1) \\ f_z(Y_2) \in (A_2 \cap B_0) \end{cases} \quad (6)$$

The more an interaction belongs to the fuzzy sets ruling a partial function (following the Gaussians laws), the more the gain of the partial function (7) will grow.

$$f_z(X_i)(x) = K_i(x, y) * x \quad (7)$$

$K_i(x, y)$ depends on the membership in the fuzzy sets defined in the matrix of decisions.

In less technical terms Whenever a subject interacts, his interaction coordinates are compared with his previous interaction coordinates and with the coordinates of the weighted mean of his five previous interactions. The Gaussian functions of Figure 2 are applied to determine the degree of membership of the mobile mean and the last interaction in near, medium, and far fuzzy sets. We apply the rules of the matrix of decisions to obtain the partial f_z functions. Finally, we sum and multiply the partial functions and their gains to obtain the final function that drives the movement of the collision boxes.

Adding Some Constraints

In theory, this algorithm can make the collision boxes drift indefinitely. Previous research has shown that the spatial perception of users can drift but can also suddenly return to a previous state. Our system might not be prepared for such a situation. To prevent this, we tested two types of constraint systems.

- A binary constraint algorithm, preventing the algorithm from placing the collision boxes further than a certain value.
- A linear constraint algorithm, diminishing the action of the algorithm past a certain value. The linear constraint decreases a gain K from 1 to 0 when the collision box is moved away from 50% to 100% of its size.

The algorithm should also prevent collision box overlap. In our algorithm, all collision boxes move the same distance at the same time. It prevents overlap and it allows the system to use data from all the buttons and be efficient faster.

Testing Through Resimulation

The data of 30 subjects touching a series of buttons, collected from previous experiments, was used to evaluate and adjust the algorithms. We ran simulations where the algorithms were applied to their interactions, with specific strategies and sets of parameters. Running these operations before an actual experiment allowed the reproduction of what subjects did thousands of times, to test hundreds of sets of parameters. Two indicators were considered, the relative error of the interaction (between −50 and +50) and the number of missed interactions.

- Linear filtering, as simple as it is, already provides decent results. For $K \in [0.2, 0.4]$, the relative error of interactions is diminished by more than 10%. Nevertheless, there are two major drawbacks.
  - It does not take into account that no user accurately hits the center every time. It overreacts when the user misses the center from time to time.
  - It does not keep any data about previous interactions, it does not learn from repetition.
- Quadratic filtering is a slight enhancement, but it inherits the same drawbacks as linear filtering. The best performance was achieved with $(K, a) = [1.2, 0.4]$.
- Weighted mobile means provide better results. After adjusting the parameters, the relative error is reduced by almost 15%. We adjusted not only the size of the mobile mean, but also the weight of each iteration. We observed that 4 or 5 iterations provides the best results, and the optimal weights are {3, 2.5, 2.5, 2.5, 2.5} for the iterations $\{n-1, n-2, n-3, n-4, n-5\}$. However, there are still many missed interactions.
- Fuzzy logic provides the best results with relative errors decreased by nearly 25% in simulations. Figure 3 shows the repartition of interactions on the buttons, with and without the algorithm. A lot of optimizations were required to achieve this result

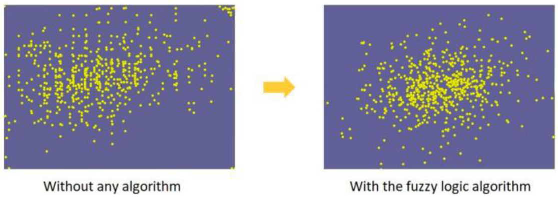

Figure 3. Repartition of interactions with and without optimized fuzzy logic algorithm. The point cloud is tighter on the right picture. The interactions failed by subjects during the recording could not be tested with the algorithm (as they were not recorded), but it is possible they could have been succeeded with it.

and the following values were found for the K_i gains.

$$\begin{cases} K_0 = 0.215 \\ K_1 = 0.41 \\ K_2 = 0.69 \end{cases}$$

These values may be too precise for us, as too much optimization would make the algorithm data dependent. Therefore, it needs to be tested with new subjects.

- The binary constraint system worsened the performance, whereas the linear constraint reduces the number of errors without degrading the performance.

Thanks to their best combines results, we chose to implement the fuzzy logic algorithm, associated with the linear constraints, into a virtual HMI. This implementation aims to evaluate this method with new subjects and data.

VALIDATION OF THE ALGORITHM

The data used in resimulations is incomplete, as no coordinates were collected about failed interactions during the recordings. Thus, we cannot know yet if our system could have turned a failed interaction into a successful one. Therefore, we conducted an experiment to compare the performance of real subjects with and without the algorithm.

Materials and Methods

This experiment took place in Renault P3I (Industrial Immersive Integration Platform) CAVE, a 4-sided virtual reality room powered by ultra-short throw full HD Panasonic projectors. It provided active stereoscopy, optical tracking with A.R.T. infrared technologies, and a 3-finger tracked glove to acquire interactions. The virtual scene was displayed by Oktal SCANeR Studio[18] and placed the subjects inside a virtual car. A custom software was displayed on their dashboard touchscreen (see Figure 4). The sensory substitution was rendered on the touchscreen whose virtual buttons would change their color when touched.

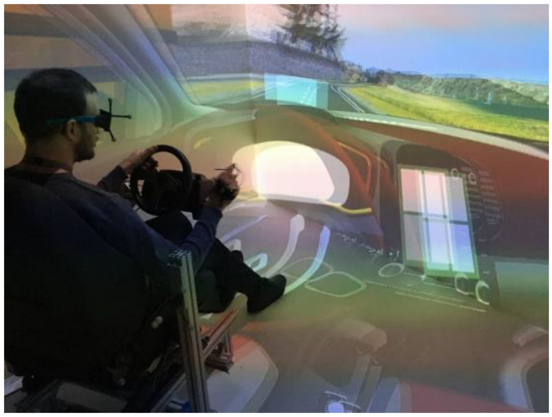

Figure 4. A subject is interacting with the virtual dashboard touchscreen.

Fifteen subjects took part in this experiment. Each one of them is a Renault employee to observe confidentiality restrictions. They were males and females, most of them between 25 and 50 years old. Their specificities were known by the questionnaires, oral questions, and verbatim records. Their feelings were collected in the end via a questionnaire.

The subjects were instructed to touch the buttons in the center as they were turned green. Each subject achieved six series of interactions: three different scales, with and without the algorithm, in a random order. Altogether, they achieved 120 interactions. A first series was always proceeded with the largest scale to allow the subjects to grasp their task.

A subjective calibration was used to initialize the space of the tracking system and to reach an approximative setup. It allowed subjects to succeed in their first interactions, hence giving initialization data to the filtering algorithm. The subjective calibration consists of three interactions on three corners of the screen. The tracking system can then map

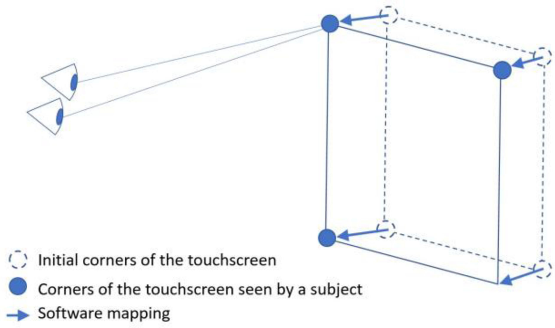

Figure 5. The collision coordinates of the dashboard touchscreen is mapped with the perceived location of the image during the initial subjective calibration.

the calculated plane with the collision coordinates of the dashboard touchscreen, as represented in Figure 5.

Results

We measured the relative error of subjects, meaning the distance between the center of the virtual button and the interaction of the subject, relative to the size of the button. The relative error is in the range of -50 to 50.

Statistics The results are paired samples, as every subject manipulated the interface with and without the algorithm. According to the Shapiro-Wilk Test, the repartition of the data is normal. We hence used the Student's *T*-test for paired samples to evaluate the degree of significance of our results and obtained a *p*-value inferior to 0.001 ($t - \text{value} = 4.56$, $df = 14$).

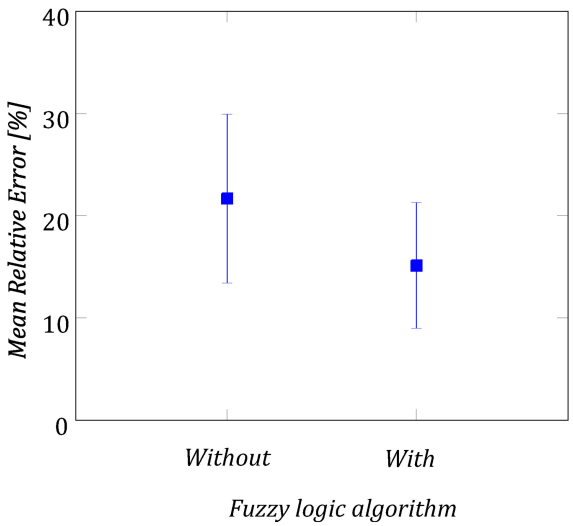

Figure 6. Mean relative error of subjects without and with the fuzzy logic algorithm. The error bars represent the standard deviation.

Table 1. Content of the questionnaire.

Question	Text
1	Did you have difficulty interacting?
2	Were you hampered while interacting?
3	Do you think you succeeded touching the center of the buttons?
4	Were you as fast as you wanted?
5	Did your vision and your interactions feel spatially consistent?
6	How much delay did you experience between your actions and expected outcomes?
7	How proficient in interacting did you feel at the end of the experience?

Relative error Relative error is represented in Figure 6. According to these results, interactions are significantly more accurate with the filtering algorithm. Indeed, the mean relative error of the subjects was reduced by 30% on average, even more than how much the simulations predicted. The filtering system allowed subjects to succeed some interactions that they would have failed otherwise by touching outside of the original collision box.

Static error During the experiment, the operator could see both the haptic and visual workspaces. He observed that most subjects encountered a heavy constant error that allowed the algorithm to be this effective. However, the algorithm needed a few (3 to 5) interactions before being totally operative.

Satisfaction of subjects The subjects answered a questionnaire at the end of the experiment. This questionnaire, built for the purpose of the experiment, contained the questions of Table 1. Every question received two answers, corresponding to filtered/ not filtered situations.

According to this questionnaire, all subjects felt more confident in their interactions with the filtering algorithm, although half of them reported that they had trouble evaluating the depth of the touchscreen. Indeed, this system improved their performance but not their visual perception. Some of them reported that they would have appreciated a tangible haptic feedback to get a stronger depth mark.

DISCUSSION AND IMPLICATIONS

These results show that such a method is relevant to improve the performance of interactions in CAVEs. However, the strategy and set of parameters used are specific to the use-case, although the fuzzy logic strategy is flexible and adaptable to many situations.

The level of performance attained is sufficient for HMI engineering, and further implementations will focus more on the industrial aspects of the simulation (repeatability, time consumption, etc.).

Drift of Spatial Perception

The hypothesis of spatial perception drift is confirmed by the fact that this algorithm worked successfully. The data is in accordance with this assumption, as it shows that each user encountered a floating drift, slightly changing throughout the experiment. However, we cannot be certain to know all the reasons for this. We know that there is an actual offset due to morphologic disparities and system imperfections, but there are other sources, like the uncertainty drift discussed earlier. The floating perception proves that we do not understand everything that happens in virtual environments as of yet.

What if it Fails

Moving the collision boxes of the buttons is not without drawbacks. If some subject drifts too much and suddenly recalibrates his perception, he or she may fail his next interaction. The constraints factor added in the algorithm prevents the subjects from drifting too far, and thus prevent them from failing interactions if their perception goes back to a previous state. However, a quick recalibration method should be implemented to secure any remaining failures. It can use gesture recognition to detect whenever users are attempting (and failing) to interact and offer them to proceed with a fast subjective recalibration. The recalibration would involve a few interactions on a dedicated 3-D interface for the algorithm to be operative again.

Generalization

This method is specific to pressing virtual buttons on a flat surface and is not ready for another use. However, using fuzzy logic (or even simpler algorithms) to filter the interactions of the users should be generalized. Further studies will focus on finding generic sets of parameters that may allow the method to be compatible with more use-cases. More issues will also need dedicated studies:

- How can the filtering data collected from a button in one position and orientation can be used for a second button in another position and orientation?
- How can the filtering include the third dimension (depth)?
- How can the method take the position and orientation of the head in the account? How can eye tracking improve our issues?

These issues are difficult to handle for now, as the system does not tackle the root causes. Instead of moving the collision engine which is supposed to be accurate, it should move the images that actually are the real issues.

To achieve this, a similar algorithm could move the virtual eyes instead of the collision engine, following the equations of Figures 7 and 8. However, to prevent users from becoming completely lost, the environment must not move after each interaction, therefore this kind of algorithm needs to make its move only rarely. Each interaction can add a contribution to the location of the eyes, but the system cannot proceed with the correction too often, thereby making real-time filtering impossible.

Nevertheless, such an algorithm may serve the calibration purpose, acting as a visual calibration system for any simulation involving CAVE-like technologies and interactions. It can be fast, user-friendly and without any storing of confidential data (like morphology). Users would only need to proceed with a few specific interactions and the algorithm would interpret where their eyes are, based on geometry formulas related to independent degrees of freedoms of the eyes (eye height, eye depth, interpupillary distance).

It could be used simultaneously with a real-time filtering to reduce the dynamic errors, as it

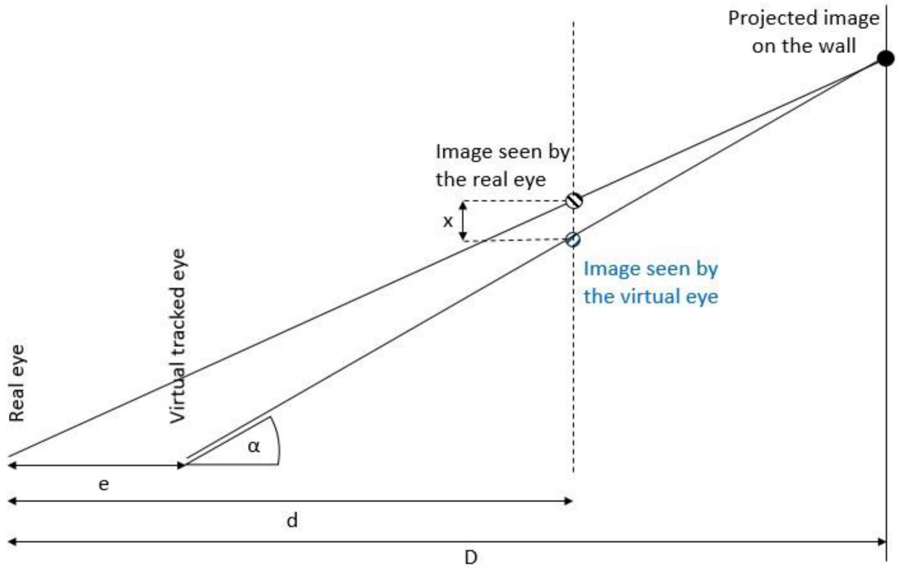

Figure 7. Quantitative influence of eye depth difference. The relation between the measured offset and the eye depth is $x = \frac{(D-e-d)}{D-e} * \frac{2*e^2*a}{e^2 + a^2 - (\frac{D-e}{\cos\alpha})^2}$.

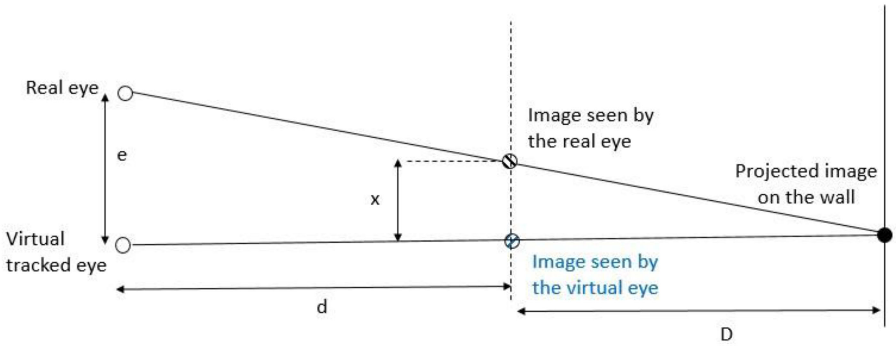

Figure 8. Quantitative influence of eye height difference. The relation between the measured offset and the eye height is $x = e * \frac{d}{d+D}$.

would only provide a correction to static ones. Of course, we need further studies to evaluate a protocol that can set-up such a calibration with a sufficient accuracy.

CONCLUSION

In this study, we tried to filter subjects' interactions to solve the mismatch between the haptic and the visual workspace in CAVEs. We aimed to improve the performance of interactions for HMI virtual engineering. Instead of moving the visual side of the simulation, we built an algorithm that moves the collision boxes of the buttons depending on where the subjects seem to see the buttons. After testing different strategies, we implemented a fuzzy logic algorithm for an actual experiment with 15 subjects and it provided significant performance improvements to their interactions. Our findings offer a better understanding of the nature of visual perception drift and solutions to counterbalance it.

Perspectives

Other studies are needed to make this concept generalizable to more interfaces. We plan to build generic sets of parameters and to make the system handle the whole virtual workspace.

These results will lead to the design of a similar algorithm to implement a low-level calibration: by moving directly the virtual eyes instead of the haptic work field, we plan to reduce the systematic error.

ACKNOWLEDGMENTS

This work was supported by a cooperation between Renault Group and Arts et Métiers Engineering School. We would like to thank in particular Renault Group for providing the materials and software we used in this experiment.

REFERENCES

1. K. Salisbury, F. Conti, and F. Barbagli, "Survey - Haptic rendering: Introductory concepts," *IEEE Comput. Graph. Appl.*, vol. 24, no. 2, pp. 24–32, Mar. 2004.
2. B. Hannaford and L. Wood, "Performance evaluation of a 6 axis high fidelity generalized force reflecting teleoperator," in *Proc. NASA Conf. Space Telerobot.*, 1989, pp. 87–96.
3. P. Galambos, "Vibrotactile feedback for haptics and telemanipulation: Survey, concept and experiment," *Acta Polytechnica Hungarica*, vol. 9, no. 1, pp. 41–65, 2012.
4. M. Kitagawa, D. Dokko, A. M. Okamura, B. T. Bethea, and D. D. Yuh, "Effect of sensory substitution on suture manipulation forces for surgical teleoperation," *Studies Health Technol. Informat.*, vol. 98, pp. 157–163, 2004.
5. R. V. Kenyon, M. Phenany, D. Sandin, and T. Defanti, "Accommodation and size-constancy of virtual objects," *Ann. Biomed. Eng.*, vol. 36, no. 2, pp. 342–348, 2008.
6. A. Lassagne, A. Kemeny, J. Posselt, and F. Merienne, "Performance evaluation of passive haptic feedback for tactile HMI design in CAVEs," *IEEE Trans. Haptics*, vol. 11, no. 1, pp. 119–127, Jan.–Mar. 2018.
7. ——, "Comparing tangible and fully virtual haptic systems for HMI studies in simulated driving situations," Stuttgart, Germany, Sep. 2017.
8. A. R. Richardson and D. Waller, "Interaction with an immersive virtual environment corrects users' distance estimates," *Hum. Factors, J. Hum. Factors Ergonom. Soc.*, vol. 49, no. 3, pp. 507–517, 2007.
9. A. Sutcliffe, B. Gault, T. Fernando, and K. Tan, "Investigating interaction in CAVE virtual environments," *ACM Trans. Comput. Hum. Interact.*, vol. 13, no. 2, pp. 235–267, 2006.
10. L. Dominjon, A. Lécuyer, J.-M. Burkhardt, and S. Richir, "A comparison of three techniques to interact in large virtual environments using haptic devices with limited workspace," in *Advances in Computer Graphics*. Berlin, Germany: Springer, 2006, vol. 4035, pp. 288–299.
11. F. Pavani, C. Spence, and J. Driver, "Visual capture of touch: Out-of-the-body experiences with rubber gloves," *Psychol. Sci.*, vol. 11, no. 5, pp. 353–359, 2000.
12. B. Lenggenhager, T. Tadi, T. Metzinger, and O. Blanke, "Video ergo sum: Manipulating bodily self-consciousness," *Science*, vol. 317, no. 5841, pp. 1096–1099, Aug. 2007.
13. F. Wang and X. Ren, "Empirical evaluation for finger input properties in multi-touch interaction," *ACM Press*, pp. 1063–1072, 2009.
14. M. O. Ernst and H. H. Bülthoff, "Merging the senses into a robust percept," *Trends Cogn. Sci.*, vol. 8, no. 4, pp. 162–169, Apr. 2004.
15. G. Klir and B. Yuan, *Fuzzy Sets and fuzzy Logic*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1995, vol. 4.
16. K. J. Aström and T. Hägglund, *PID Controllers: Theory, Design, and Tuning*. Pittsburgh, PA, USA: Isa Research Triangle Park, 1995, vol. 2.
17. R. E. Kalman and others, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, 1960.
18. A. Kemeny, "A cooperative driving simulator," in *Proc. Int. Train. Equip. Conf.*, 1993, pp. 67–71.

Antoine Lassagne is currently working toward the Ph.D. degree at the Laboratory of Immersive Visualization and at Renault Group, Boulogne-Billancourt, France. He is interested in haptic rendering and virtual reality. He received the Master's in engineering from Arts et Métiers, Paris, France, in 2015. Contact him at Antoine.lassagne@renault.com.

Andras Kemeny is the Director of the Laboratory of Immersive Visualization, a joint Renault - Arts et Métiers ParisTech research laboratory. He has initiated and directed the development of a driving simulation software package, SCANeR Studio, a worldwide leader and a major development and assessment driver in the loop simulation tool for automotive driver assistance systems, human machine interfaces, autonomous and connected vehicles, deployed at a large number of automotive OEMs, suppliers, and research laboratories. He is also Associate Professor with Arts et Métiers ParisTech and author of more than 150 scientific and technical papers and book chapters as well as several industrial patents and software copyrights. Contact him at Andras.kemeny@ensam.eu.

Javier Posselt is member of Renault's Autonomous Driving Simulation & Virtual Reality department since 2010 as Project Manager in virtual and mixed reality systems. For 15 years, he has been Project Manager on projects involving industrial process simulation (virtual factory), and digital mockup assessment using virtual and mixed reality systems. He managed French national and international collaborative projects in the area of virtual reality. He was Project Manager of the first high quality projection CAVE of Renault in operation since 2013. He received the Ph.D. degree in robotics from Université Pierre et Marie Curie, Paris, France, in 1996. Contact him at Javier.posselt@renault.com.

Frédéric Merienne is currently a Professor with Arts et Métiers and Director of the Image Institute, a research team of the Le2i Laboratory, since 2004. His research interests include virtual immersion linked with engineering, cultural heritage, and health applications. He is the author of many scientific papers in virtual reality and related disciplines. He is involved in different projects with industrial partners and initiated international collaborative projects in the area of virtual reality with universities in the USA, Australia, Colombia, and Malaysia. He received the Ph.D. degree from the National Polytechnical Institute of Grenoble, Grenoble, France, in 1996. Contact him at Frederick.merienne@ensam.eu.